基于改进正余弦优化算法的多阈值图像分割 *

郎春博，贾鹤鸣†，邢致恺，彭晓旭，李金夺，康立飞

(东北林业大学 机电工程学院，哈尔滨 150040)

摘　要：针对多阈值图像分割方法计算量大、分割精度低的问题，提出了基于改进正余弦算法（improved sine cosine algorithm）的多阈值图像分割方法。首先对种群进行混沌初始化来提高初始种群质量；其次根据粒子适应度值的大小自适应地调整参数；最后引入反向学习策略并择优选取粒子。伯克利图像和植物冠层图像分割实验的结果表明，该算法的运行时间较短，而且分割精度较高，具有较强的鲁棒性。

关键词：正余弦算法；多阈值图像分割；混沌初始化；自适应；反向学习

中图分类号：TP391.41　　doi: 10.19734/j.issn.1001-3695.2018.10.0779

Multi-threshold image segmentation based on improved sine cosine optimization algorithm

Lang Chunbo, Jia Heming†, Xing Zhikai, Peng Xiaoxu, Li Jinduo, Kang Lifei

(*College of Mechanical & Electrical Engineering*, *Northeast Forestry University*, *Harbin* 150040, *China*)

Abstract: Aiming at the problem of the computational complexity and low segmentation precision of the multi-threshold image segmentation method, this paper proposed an Improved Sine Cosine Algorithm (ISCA) based multi-threshold image segmentation method. Firstly, this method used a chaotic initialization technique to improve the quality of initial population. Secondly, it introduced an adaptive strategy to adjust the parameters according to the fitness values. Finally, it utilized an opposition-based learning strategy, then the better particles were selected. The results of the Berkeley image and the plant canopy image segmentation experiments show that this method has a satisfied performance in terms of running time and segmentation accuracy. And it has a strong robustness.

Key words: sine cosine algorithm; multi-threshold image segmentation; chaos initialization; adaptive; opposition-based learning

0　引言

图像分割是由图像处理到图像分析的关键步骤，其目的在于将图像分割成若干个独立的、具有特殊性质的部分并提取出感兴趣目标。目前常用的图像分割方法有基于区域的分割方法、基于边缘的分割方法、基于特定理论的分割方法以及基于阈值的分割方法等[1~4]。而阈值分割方法因其计算量小、易于实现、效率高、性能稳定等优点被广泛应用于很多领域。例如，在交通图像分析中，汽车牌照的识别与提取[5]；在医学图像分析中，PET-MR 图像判断病情的恶化或好转[6]；在林学图像分析中，研究森林火灾图像来检测实时火情，为实现火灾的预测预报奠定坚实基础[7]。传统的单阈值分割方法适应性并不广泛，因此一些学者利用迭代的方式将其扩展至多阈值。

多阈值图像分割可以被看成一种优化问题。传统多阈值方法的本质就是穷举法，其计算量会随着阈值个数的增加呈指数增长，实时性较差。为此有学者利用智能算法对分割的适应度函数进行寻优操作，在一定程度上解决了运行效率的问题[8~11]。但由于算法本身的局限性，寻优能力并不理想，易早熟收敛、陷入局部最优等状况仍需得到有效地改善。

正余弦算法（sine cosine algorithm, SCA）是由 Mirjalili 等人[12]于 2016 年提出的新颖随机群优化算法。该算法的核心思想是利用正余弦函数的震荡特性，向外波动进行全局探索，向最优解波动进行局部开发，使算法逐步收敛。SCA 具有易于实现，收敛速度快，求解精度高等优点，其随机参数与递减参数的设置也使得算法的探索与开发能力得到很好的平衡，但局部搜索能力较弱的缺点使得算法效果并不理想。

本文提出一种改进正余弦算法（improved sine cosine algorithm, ISCA）并将其应用到多阈值图像分割领域。为解决标准 SCA 局部搜索能力较弱的问题，在保证运行效率的同时提高分割精度，进行了以下几部分改进：a) 混沌初始化种群，提高初始种群质量;b) 采用一种自适应参数策略代替原有的线性递减参数策略，根据粒子适应度值的大小动态调整参数以更好地平衡探索与开发能力;c) 引入反向学习策略增加可选粒子数量，并择优选取粒子，增大随机性的同时提高种群进化速度。实验选取四幅伯克利图像以及两幅植物冠层图像，对比 SAPSO、SSA、PSO、ALO 与 HSO 五种优化算法。实验结果表明，ISCA 增强了算法的局部搜索能力，使得求解精度得到有效地提高，其运行时间、分割精度等指标均优于对比算法，具有较高的鲁棒性，为后续的图像提取操作奠定了良好的基础。

收稿日期：2018-10-09；**修回日期：**2018-11-08　**基金项目：**中央高校基本科研业务费专项资金资助项目（2572014BB03）；国家自然科学基金资助项目（51609048）；黑龙江省研究生教育创新工程项目（JGXM_HLJ_2016014）

作者简介：郎春博(1998-)，男，辽宁沈阳人，硕士研究生，主要研究方向为智能优化与图像处理；贾鹤鸣 (1983-)，男（通信作者），黑龙江哈尔滨人，副教授，博士，主要研究方向为图像处理与信息检测技术（jiaheminglucky99@126.com）；邢致恺(1993－)，男，黑龙江大庆人，硕士研究生，主要研究方向为图像处理技术；彭晓旭(1995-)，男，黑龙江齐齐哈尔人，硕士研究生，主要研究方向为图像分割和智能优化算法；李金夺(1996-)，男，吉林桦甸人，硕士研究生，主要研究方向为神经网络优化技术；康立飞(1993-)，男，山东日照人，硕士研究生，主要研究方向为图像处理技术.

1　多阈值 Otsu 图像分割

Otsu 算法（最大类间方差法）是一种自适应确定二值化图像分割阈值的算法，由日本学者大津于 1979 年提出，该算法易于实现且在一定条件下不受亮度或对比度影响，因而被广泛应用于图像处理领域[13~15]。其核心思想为：利用阈值将图像分割为背景、前景两部分，使得这两部分类间方差最大的阈值便为最优阈值。

假设一幅图像具有 $M \times N$ 像素、L 个灰度级，其中灰度级为 i 的点的概率为

$$p_i = n_i / MN \tag{1}$$

其中：n_i 表示灰度值为 i 的像素个数。

设阈值 t 将整幅图像分为前景和背景两部分，则类间方差 σ_B^2 是 t 的函数：

$$\sigma_B^2(t) = P_0 \times (m_0 - m_G)^2 + P_1 \times (m_1 - m_G)^2 \tag{2}$$

其中：$P_0 = \sum_{i=0}^{t} p_i$，$P_1 = \sum_{i=t+1}^{L-1} p_i$

$$m_0 = \frac{1}{P_0}\sum_{i=0}^{t} ip_i \quad m_1 = \frac{1}{P_1}\sum_{i=t+1}^{L-1} ip_i \quad m_G = \sum_{i=0}^{L-1} ip_i$$

P_0 和 P_1 分别表示像素被分到前景和背景两类中的概率，m_0 和 m_1 分别表示两类的平均灰度，m_G 表示整幅图像的平均灰度，使 $\sigma_B^2(t)$ 达到最大时的 t^* 即为最优阈值。

将该公式推广至多阈值，设阈值组合为 $(t_1, t_2, \ldots, t_{K-1})$，此时图像被分为 K 个类别，类间方差表示为：

$$\sigma_B^2(t_1, t_2, \ldots, t_{K-1}) = \sum_{j=0}^{K-1} P_j (m_j - m_G)^2 \tag{3}$$

使得 $\sigma_B^2(t_1, t_2, \ldots, t_{K-1})$ 达到最大值的阈值组合 $(t_1^*, t_2^*, \ldots, t_{K-1}^*)$ 即为最佳阈值组合。但传统的多阈值 Otsu 方法其本质就是穷举法，总计算量近似为 $O(L)^K$，且随着阈值个数 K 的增加呈指数级增长，为提高程序运行效率，现采用 ISCA 算法对多阈值选取过程加以优化。

2　正余弦优化算法（SCA）

正余弦优化算法是由 Mirjalili 等人于 2016 年提出的新的随机群优化算法，其基本思想是利用正余弦函数的震荡特性逐步收敛于最优解，向外波动进行全局探索，向最优解波动进行局部开发。

利用 SCA 算法求解优化问题的思路为：在 D 维空间中初始化产生数量为 m 的种群，种群中第 $i\,(i = 1, 2, \ldots, m)$ 个粒子的位置表示为 $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$，通过计算粒子对应适应度函数值 $f(x)$ 并加以比较，得到当前最优个体位置 $p_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$，其中粒子更新位置的公式如下：

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t + r_1 \times \sin(r_2) \times |r_3 p_{gj}^t - x_{ij}^t| & r_4 < 0.5 \\ x_{ij}^t + r_1 \times \cos(r_2) \times |r_3 p_{gj}^t - x_{ij}^t| & r_4 \geq 0.5 \end{cases} \tag{4}$$

其中：$r_2 \in (0, 2\pi)$，$r_3 \in (0, 2)$，$r_4 \in (0, 1)$ 为均匀随机数，参数 r_1 线性递减，其表达式如下：

$$r_1 = a \times \left(1 - \frac{t}{t_{max}}\right) \tag{5}$$

其中：a 为常数，t 为当前迭代次数，t_{max} 为最大迭代次数。由式(4)可以看出 SCA 算法共有 4 个主要参数，分别为 r_1, r_2, r_3 和 r_4。正是因为随机参数和线性递减参数的设置使得算法很好地平衡了全局探索与局部开发能力[16]。参数 r_1 决定了粒子下一次移动的方向，参数 r_2 则控制了移动的距离，参数 r_3 引入了一定的修正策略，赋给当前最优解一个随机的权值，以增大或减小当前移动方向所产生的效果，参数 r_4 的作用是在正余弦函数间随机切换。

3　改进正余弦优化算法（ISCA）

3.1　混沌初始化

初始种群质量对 SCA 算法来说是至关重要的，而一个良好的初始种群可使其在收敛速度、求解精度等方面得到显著提升。以往的初始化操作是基于 rand 函数进行的，尽管该方法有着良好的随机性，但其分布情况往往不尽人意。而混沌变量具有随机性、遍历性、规律性的特点，可以很好地保持群体多样性[17~18]。这里选用 Logistic 这一经典混沌映射进行初始化操作，其表达式如下：

$$x_{n+1} = \mu x_n (1 - x_n) \tag{6}$$

其中：$\mu \in (0, 4]$，$x_n \in (0, 1)$，μ 越大混沌性越高，$\mu = 4$ 时系统处于完全混沌状态。

混沌初始化步骤如下：

a)利用 rand 函数随机生成种群的第一个粒子 $x(i,:)$。

St 混沌初始化下一个粒子，对 $x(i,:)$ 的每个维度进行归一化操作并赋值给 $x(i+1,:)$，使其在 $[0,1]$ 内。

c)根据式（6）进行 Logistic 映射，生成混沌变量 $cx(j)$，其中 $j = 1, 2, \ldots, D$，代表当前维度。

d)利用混沌变量 $cx(j)$ 将 $x(i+1, j)$ 映射回原始空间，公式为

$$x(i+1, j) = x_{min} + cx(j) * (x_{max} - x_{min})。$$

e)判断是否遍历所有粒子，否则转向 b)。

3.2　自适应参数策略

根据以上分析不难看出，SCA 算法中参数 r_1 起着至关重要的作用，当 $r_1 > 1$ 时控制粒子向远离最优解方向移动以加强全局搜索能力，$r_1 < 1$ 时控制粒子向着最优解移动以加强局部开发能力。但标准 SCA 算法中参数 r_1 仅仅随着迭代次数的增加而线性减小，并没有考虑到每个粒子的实际情况，严重影响了种群的进化速度以及算法寻优能力。

因此本文采用了一种自适应参数调整策略，该策略可以针对每个粒子的不同情况加以赋值，在保证全局探索质量的同时增强了 SCA 算法局部开发能力[19]。相较原来仅随迭代次数变化而线性改变参数的策略而言，自适应参数策略普遍性更强、适应度更高，公式如下：

$$r_1 = \begin{cases} r_{1med} - \left|\dfrac{f_i - f_{bav}}{f_g - f_{bav}}\right| \times (r_{1med} - r_{1min}) & f_i < f_{bav} \\ (r_{1max} - r_{1min}) \times \left(1 - \dfrac{t}{t_{max}}\right) & f_{bav} \leq f_i \leq f_{av} \\ r_{1med} + \dfrac{(r_{1max} - r_{1min})}{2} \times \text{rand} & f_i > f_{av} \end{cases} \tag{7}$$

其中：f_i 为第 i 个粒子的适应度函数值；f_{av} 为所有粒子的平均适应度函数值；f_{bav} 为所有适应度值优于 f_{av} 的粒子其适应度值的平均；f_g 为当前最优适应度值；r_{1med} 的定义如下：

$$r_{1med} = \frac{r_{1max} + r_{1min}}{2} \tag{8}$$

当 $f_i < f_{bav}$ 时，当前粒子距离种群中最优粒子位置较近，应赋给其一个较小的参数 r_1 来加强算法的局部开发能力，避免错过最优位置；当 $f_{bav} \leq f_i \leq f_{av}$ 时，当前粒子在种群内质量一般，令其参数 r_1 随迭代次数增加而线性递减以较好的平衡探索与开发能力；当 $f_i > f_{av}$ 时，当前粒子距离最优粒子较远，位置较不理想，故赋给其一个较大的参数 r_1 以尽快向最优粒子靠近。

3.3 反向学习策略

反向学习策略是由 Tizhoosh 于 2005 年提出的一种优化学习方法，该策略的理论依据是：每个随机产生的候选解与它的反向解相比有 50%的概率更远离最优解[20]。因此，通过寻找当前解的反向解，并对二者进行评估，贪婪地选择较优秀的解作为候选解，进而引导种群向最优解逼近，提高种群进化速度以及算法的寻优能力，公式如下：

$$x_{ij}^{*}=k\left(a_j+b_j\right)-x_{ij} \tag{9}$$

其中：x_{ij}^{*} 为种群中第 i 个粒子在第 j 维度上的反向数；a_j 和 b_j 分别为第 j 维度的上界与下界；参数 k 决定反向学习的模型。

本文依据一定概率 p 执行反向学习操作，旨在增大随机性的同时降低算法复杂度。

3.4 算法描述

混沌初始化策略能够生成高质量的初始种群，增强粒子分布的随机性；自适应参数策略则提高了粒子与其对应参数 r_1 的适配度，与仅随迭代次数增大而改变的线性递减策略相比，更好地平衡了 SCA 算法的探索与开发能力；反向学习策略增加了种群多样性，贪婪地选取优秀个体，增强算法局部开发能力的同时加快了种群进化速度，提高搜索效率，其流程如图 1 所示。

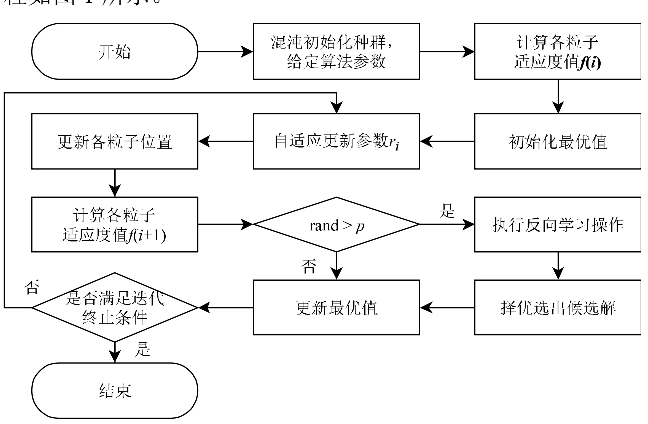


图 1　改进正余弦算法流程图

Fig. 1　Flow chart of the Improved Sine Cosine Algorithm

改进正余弦算法（ISCA）的基本步骤如下：

a)随机生成群体规模为 m 初始种群，利用式（6）混沌初始化每个粒子的位置，预先设定算法执行所需参数。

b)计算出每个粒子适应度值 $f(i)$，并从中选出最优值记为 f_g。

c)根据式（7）自适应地调整每个粒子的参数 r_1。

d)根据式（4）更新各粒子位置，同时将其限制在最大范围 $[0, L-1]$ 内。

e)将更新后的位置代入适应度函数得到新的适应度值 $f(i+1)$。

f)若 $\text{rand}>p$，则根据式（9）进行反向学习操作，并在当前解与反向解中择优选出候选解。

g)将候选解对应适应度值与最优值 f_g 比较，若优于 f_g 则将其更新。

h)判断是否满足迭代终止条件，若满足则退出循环，否则转向 c)。

4 实验结果及分析

4.1 实验背景

为了验证 ISCA 算法在多阈值图像分割中的有效性，选取4幅伯克利经典图像以及2幅植物冠层图像加以分析研究，并与 SAPSO、SSA、PSO、ALO、HSO 算法进行比较。其中 SAPSO 算法引入模拟退火思想，利用模拟退火算法的“突跳”能力有效地避免陷入局部最优，提高了算法性能；SSA 樽海鞘算法是于 2017 年提出的新颖优化算法；ALO 蚁狮算法与 HSO 和声搜索则在近些年被广泛应用于优化领域。本文通过程序运行时间以及图像分割精度来对各算法性能进行评判。其中对于图像分割精度的分析引入峰值信噪比（PSNR）和结构相似性（SSIM）两个指标，其定义如下[21]：

$$PSNR=10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{10}$$

$$MSE=\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I(i,j)-K(i,j)\right]^2 \tag{11}$$

其中：$I(i,j)$ 和 $K(i,j)$ 分别表示大小为 $M\times N$ 的原图像和分割图像。

$$SSIM(x,y)=\frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \tag{12}$$

其中：μ_x 和 μ_y 分别为原图像和分割图像平均灰度；σ_x^2 和 σ_y^2 分别为原图像和分割图像的方差；σ_{xy} 为原图像和分割图像的协方差；$c_1=6.5025$，$c_2=58.5225$

植物冠层图像采集自东北林业大学模式识别与智能系统实验室，其图像分辨率为 3000×4000。算法的实验环境为 Windows10 系统，1.6 GHz 处理器，8 GB 内存，64 位操作系统，编程环境为 MATLAB R2017a。实验中各参数设置如下：粒子数目 $n=30$，最大迭代次数 $t_{\max}=500$，各算法特性参数设置如表 1 所示。

表 1　各算法参数设置

Table 1　Parameters of each algorithm

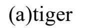
算法	参数	取值
ISCA	r_1	[0,2]
	p	0.5
	μ	4
SAPSO	T_0	-g/ln(0.1)
	k	0.95
SSA	c_1	[0,2]
PSO	ω	[0.4,0.9]
	$v_{\max}$	25.5
	c_1	2
	c_2	2
ALO	w	[2,6]
HSO	PAR	0.3
	$HCMR$	0.95
	BW	25.5

4.2 伯克利经典图像分割实验

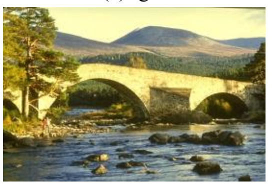
为了验证本文优化算法在多阈值图像分割中的普适性，在伯克利图像库中随机选取 4 幅图片，如图 2 所示。以类间方差函数作为算法的适应度函数，利用 ISCA 算法对以上 4 幅图像多阈值分割的阈值选取过程加以优化。实验中所有图片选取的阈值为 2、3、4、5。为定量地比较各优化算法在多阈值图像分割中的性能，本文对程序运行时间以及图像分割精度两方面进行了研究，其中各算法阈值情况如表 2 所示。首先对各算法分割阈值加以分析：当 K=2 时，所有算法的取值几乎相同。当 K=3 时，ISCA 算法搜索空间扩大，全局搜索能力得到提高，而其他算法性能提高不明显。当 K=4，5

时，ISCA 算法的全局搜索能力优势得以进一步体现，可在整个解空间内进行随机搜索操作，有效地避免了陷入局部最优。

其次对各算法运行时间加以分析：由表 3 中数据可知，随着阈值个数的增加，算法复杂度呈指数次幂增长趋势，运行时间也随之增加。阈值个数 K=2，3 时，各算法运行时间差距较小。当取到较高阈值个数 K=5 时，各算法运行时间差距较大，其中 ALO 算法时间最慢。当 K=3,4 时 PSO 算法与 ISCA 算法运行时间相近。但当阈值个数提升到 5 时，本文算法的优势得以体现，其运行时间随阈值个数增加而变化的幅度较小。本文提出的 ISCA 算法，尽管随着阈值个数的增加运行时间会有所增加，但在阈值个数 K=5 时仍运算时间优于其他算法，说明本文算法效率高，能够较快地完成图像分割任务。



(a)tiger


(b)horseman


(c)bridge


(d)pig

图 2　伯克利原始图像

Fig. 2　Original Berkeley images

表 2　各算法最佳分割阈值

Table 2　Optimal threshold values obtained by each algorithm

图像	K	ISCA			SAPSO			SSA			PSO			ALO			HSO		
		R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
tiger	2	75	84	69	78	83	66	77	83	66	77	83	66	77	83	66	77	83	66
		139	150	119	146	149	125	145	149	125	145	149	125	145	149	125	146	148	125
	3	67	76	48	59	69	56	59	69	56	59	69	56	59	69	56	59	69	56
		104	111	70	105	105	85	105	105	85	105	105	85	105	105	85	105	106	85
		168	166	141	170	165	140	170	165	140	170	165	140	170	165	140	170	168	140
	4	53	58	49	53	60	51	53	60	51	53	60	51	53	60	51	51	61	51
		86	87	69	88	87	74	88	87	74	88	87	74	88	87	74	81	89	77
		125	107	126	132	122	107	132	122	107	132	122	108	132	122	107	125	122	111
		197	154	208	191	179	163	191	179	163	191	179	164	191	179	163	191	179	167
	5	43	31	49	47	56	46	47	57	46	48	56	46	47	56	46	48	59	45
		53	54	64	73	80	65	73	81	64	74	80	64	73	80	65	76	84	64
		78	78	93	105	107	88	105	108	86	106	107	86	105	107	87	109	111	87
		114	123	112	145	144	124	145	145	121	146	144	120	145	144	122	148	154	132
		167	168	157	199	197	181	199	198	177	200	197	176	199	197	178	198	204	196
horseman	2	92	76	59	90	77	61	90	77	61	90	77	61	90	77	61	90	77	61
		180	160	142	174	158	136	174	158	136	174	158	136	174	158	136	174	157	137
	3	70	53	27	63	59	34	63	59	35	63	59	35	63	59	35	63	60	33
		121	104	65	114	104	74	114	104	75	114	104	75	114	104	75	114	105	73
		185	167	121	186	172	140	186	172	140	186	172	140	186	172	140	186	173	141
	4	66	55	16	60	53	30	60	54	30	60	53	30	60	54	30	60	53	32
		109	93	66	106	93	65	106	94	65	106	93	65	106	94	65	106	94	67
		177	150	103	158	139	99	158	140	99	158	139	99	158	140	99	160	141	102
		212	206	145	213	199	149	213	199	149	213	199	149	213	199	149	215	204	152
	5	39	41	22	52	42	32	52	42	28	52	42	28	52	42	28	51	43	29
		72	72	35	84	72	66	84	72	59	84	72	59	84	72	59	82	76	60
		115	102	70	121	105	100	121	105	89	121	105	89	121	105	89	118	107	89
		155	134	132	166	146	131	166	146	123	166	146	123	166	146	123	163	149	123
		200	200	160	217	203	176	217	203	170	217	203	170	217	203	170	215	207	169
bridge	2	100	106	72	104	104	67	104	104	67	104	104	67	104	104	67	104	103	67
		187	188	135	190	191	137	190	191	137	190	191	137	190	191	137	190	190	138
	3	95	84	60	88	85	46	88	85	46	88	85	46	88	85	46	87	85	46
		154	142	94	143	137	100	143	137	100	143	137	100	143	137	100	142	137	100
		210	200	149	210	204	153	210	204	153	210	204	153	210	204	153	209	204	153
	4	62	78	42	73	69	35	73	69	35	73	69	35	73	69	35	75	68	36
		113	123	94	113	109	75	113	109	75	113	109	76	113	109	75	116	109	74
		156	143	141	158	151	116	158	151	116	158	151	117	158	151	116	161	151	116
		208	195	175	216	210	159	216	210	159	216	210	159	216	210	159	219	211	159
	5	63	79	5	66	64	30	66	64	29	66	64	35	66	64	29	69	65	31
		95	113	51	100	99	63	100	98	62	100	99	75	99	99	62	104	98	65
		147	129	85	134	132	99	134	131	99	134	132	116	133	132	99	136	135	101
		186	165	128	173	166	133	173	165	133	173	166	142	172	166	133	174	172	136
		231	216	157	222	216	166	222	216	166	222	216	170	222	216	166	224	215	164
pig	2	69	58	40	70	61	37	70	61	37	70	61	37	70	61	37	70	61	37
		145	117	93	149	122	88	149	122	88	149	122	88	149	122	88	149	122	89
	3	57	51	28	55	46	37	55	46	28	55	46	28	55	46	28	54	43	29
		106	74	57	97	81	88	97	81	54	97	80	54	97	81	54	96	79	55
		167	144	95	162	132	124	162	132	97	162	131	97	162	132	97	162	132	99
	4	63	46	17	46	40	25	46	40	25	46	40	25	46	40	25	47	41	23
		91	75	38	82	71	47	82	71	47	82	71	47	82	71	47	82	72	44
		116	85	66	123	104	75	123	104	75	124	104	75	123	104	75	123	106	72
		189	127	108	180	148	114	180	148	114	181	148	114	180	148	114	180	151	109
	5	36	45	25	38	35	23	38	35	23	39	35	23	38	35	23	37	36	22
		55	74	50	68	60	42	68	60	42	68	61	42	68	60	42	67	60	44
		95	111	69	97	84	62	98	84	62	98	85	62	98	84	62	98	85	64
		117	155	106	136	115	90	137	115	90	137	116	90	137	115	91	136	114	101
		184	218	160	188	156	126	189	156	126	189	158	126	189	157	127	191	155	135

表 3　各算法运行时间

Table 3　Running time obtained by each algorithm

图像	K	时间（s）					
		ISCA	SAPSO	SSA	PSO	ALO	HSO
tiger	2	3.0457	4.4085	5.0922	4.4982	7.6689	4.0567
	3	3.1123	4.7359	5.1177	4.5569	7.9659	5.6664
	4	3.5673	4.7697	5.2516	4.6020	8.8888	6.4161
	5	3.9151	5.3457	5.3464	4.7452	10.1538	7.2517
horseman	2	3.3988	3.5215	4.0104	3.4799	6.4452	4.1924
	3	3.6110	4.6704	5.2647	4.2391	7.6996	4.6256
	4	4.0773	4.9945	5.3161	4.6492	9.1019	4.8122
	5	4.1974	5.3839	5.8027	4.8519	10.8712	5.2971
bridge	2	3.4149	4.0749	4.0320	4.6851	7.3420	4.3063
	3	3.6381	4.6919	5.4292	4.6422	8.4288	4.9381
	4	3.7298	4.1349	5.4154	4.3438	9.3038	5.1670
	5	4.2519	4.7859	5.4232	4.8032	10.6338	5.4593
pig	2	3.2110	3.8976	5.1616	4.4941	7.0998	4.3744
	3	3.6821	4.9248	5.2334	4.6945	7.9815	4.5586
	4	3.9859	4.9545	5.3763	4.7266	9.2002	4.7509
	5	4.1130	5.2493	5.7323	4.8389	9.8534	5.4511

表 4　各算法分割图像指标

Table 4　Indices of segmented image obtained by each algorithm

图像	K	PSNR						SSIM					
		ISCA	SAPSO	SSA	PSO	ALO	HSO	ISCA	SAPSO	SSA	PSO	ALO	HSO
tiger	2	14.3932	14.3737	14.3832	14.3832	14.3832	14.1392	0.3183	0.3169	0.3181	0.3181	0.3181	0.3087
	3	17.0466	17.0313	17.0313	17.0313	17.0313	16.7344	0.4638	0.4638	0.4638	0.4530	0.4638	0.4636
	4	19.3209	19.1908	19.1908	19.1947	19.1908	19.2525	0.5737	0.5679	0.5679	0.5675	0.5679	0.5700
	5	21.6244	21.0907	20.9584	20.9790	21.0748	20.8383	0.6896	0.6522	0.6485	0.6494	0.6525	0.6398
horseman	2	14.4799	14.3832	14.3832	14.3832	14.3832	14.3776	0.3252	0.3181	0.3181	0.3181	0.3181	0.3180
	3	17.0459	17.0313	17.0313	17.0313	17.0313	16.8233	0.4638	0.4557	0.4638	0.4638	0.4638	0.4498
	4	19.7594	19.1917	19.1908	19.1878	19.1908	18.9160	0.5693	0.5675	0.5679	0.5679	0.5679	0.5498
	5	22.4564	20.9709	21.0748	21.0688	20.9601	20.7858	0.6874	0.6481	0.6525	0.6525	0.6485	0.6444
bridge	2	14.2048	14.1361	14.1361	14.1361	14.1361	14.0884	0.4934	0.4918	0.4918	0.4918	0.4918	0.4889
	3	16.7809	16.7379	16.7379	16.7379	16.7379	16.1889	0.6204	0.6191	0.6191	0.6191	0.6191	0.5887
	4	19.3742	18.9590	18.9590	18.9588	18.9590	18.8898	0.7114	0.7113	0.7113	0.7070	0.7113	0.7082
	5	20.3207	20.2488	20.2551	19.1329	20.2535	19.9317	0.7693	0.7680	0.7452	0.7637	0.7691	0.7593
pig	2	15.3413	15.3083	15.3083	15.3083	15.3083	15.3390	0.4216	0.4214	0.4214	0.4214	0.4214	0.4186
	3	18.1568	17.2892	17.2620	17.2335	17.2620	17.3977	0.5546	0.5246	0.5509	0.5510	0.5509	0.5346
	4	19.5513	19.5423	19.5423	19.5413	19.5423	19.3491	0.6394	0.6377	0.6377	0.6375	0.6377	0.6354
	5	23.8224	21.3863	21.4014	21.4419	21.5022	21.9999	0.7016	0.6994	0.6993	0.6978	0.6996	0.6916

多阈值图像分割的阈值个数越多，其分割图片的精度越高，即与原图更加接近。为定量地评估各优化算法在分割效果方面的能力，本文对各算法的 PSNR 和 SSIM 指标进行比较，其结果如表 4 所示。首先分析峰值信噪比 PSNR，当 K=2，3 时，各优化算法取值相近，说明其分割图像差别也不是很大，但相比之下本文算法仍有一定优势。当 K=4，5 时，ISCA 算法的分割指标增加明显，对于 tiger 和 horseman 这种目标与背景相差较小、不易处理的图像，数值有明显提升。说明本文算法不仅适用于多维函数极值求解问题，在图像分割领域也有很强的工程实用性。其次分析结构相似性 SSIM，从表 4 中可以看出，随着阈值个数的增加，分割图像的失真程度越低，更加接近于原图像。本文算法得到的指标均优于对比算法，ALO 算法随着阈值个数的继续增加，其结构相似性指标则低于本文算法获得指标，说明 ALO 算法所得阈值与理想分割阈值相比仍有着一定差距，进而验证了 ISCA 算法寻优精度高，能够得到较好的分割阈值。本文提出的 ISCA 算法与其他对比算法相比性能最佳，得到的分割图像与原图像相似度最高，说明该算法能够胜任复杂图像的处理任务，更好地为后续植物冠层图像的识别与分割提供支持。为进一步证明多阈值分割方法在图像处理领域中的优异性能，能很好地将图像分为若干个具有特殊性质的独立区域，在下一节中将会对植物冠层图像进行实验与分析。

4.3　植物冠层图像分割实验

植物的冠层作为植物最先接触到外界气体环境和光照的部位，其结构形态与植物的生长状况密切相关。为提高植物生长状态评估的可靠性，降低由于图像分割技术不成熟导致测算误差的可能，本节对植物冠层图像的分割进行了研究。选取 2 幅植物冠层图像进行多阈值分割实验，如图 3 所示，并与改进分水岭算法以及模糊聚类算法进行分割效果对比[22~24]。为得到较好的分割效果阈值个数以及聚类数均设置为 5。实验中 ISCA 算法参数的设置与上节一致，模糊聚类算法迭代终止条件设为相邻两次迭代差值小于 0.001。

为从分割图像中分割出植物冠层部分，本文进行了如下操作：

a）从分割得到的 K+1 个区域中选取含有植物主体的区域并将其置 1，其余部分置 0。

b）对选取的区域进行形态学处理（腐蚀，膨胀等操作），在得到的图像中提取最大连通区。上述操作反复进行，直到获得外轮廓线清晰且闭合的图像。

c）对得到的最大连通区图像进行填洞处理并与原图像矩阵点乘，即可得到分割后的植物图像如图 4 所示。

为了更好的验证本文提出的算法，将最大熵作为评测指标对比各算法的分割图像中的信息量，其定义如下：

$$Entropy = -P_0 \log P_0 - P_1 \log P_1 \tag{13}$$

其中: P_0 与 P_1 分别为图像前景、背景像素占整幅图像的概率。熵值越大代表图像所包含信息越丰富，即图像质量越高。


(a)plant1

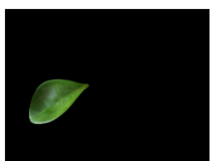

(b)plant2

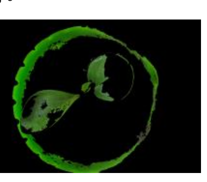
图 3　植物冠层图像

Fig. 3　Plant canopy images

各算法的植物冠层图像的分割图如图 4 所示，从图 4 可以明显地看出，本文提出的基于 ISCA 算法的多阈值图像分割方法优于其他分割算法，完整地将植物的叶片区域进行分割。从 Plant1 分割图像中可以看出，改进分水岭算法分割效果较差，存在欠分割现象，只能获得部分叶片。模糊聚类算法尽管获得叶子大致轮廓但仍存在叶片残缺及花盆区域，存在过分割现象。而相比之下本文算法分割质量较高，能够将花盆与冠层主体很好进行分割。从 Plant2 植物冠层的分割结果图可知，分水岭算法的分割能力较弱，对叶片的区域分割不完整。模糊聚类算法能够分割冠层主体区域，但由于叶片的受光不均导致分割效果较差，使叶片的分割结果存在残缺。本文算法有效地解决了受光不均的问题，很好地将冠层主体归为一类，有效的对叶片区域进行了分割。


(a)本文算法


(b)改进分水岭算法


(c)模糊聚类算法


(d)本文算法


(e)改进分水岭算法


(f)模糊聚类算法

图 4　冠层分割图像

Fig. 4　Segmented images of plant canopy

各算法分割后图像的最大熵值和运行时间如表 5 所示。从表 5 中的最大熵值可知，改进的分水岭数值最差，本文提出的算法优其他对比算法，反映图像中的信息更多，分割效果更好。从 CPU 运行时间可知，模糊聚类算法尽管在分割效果上优于改进分水岭算法，但其运行时间较长。而本文算法的程序运行时间最短，对于 K=5 这种高阈值情况其运行时间较短，说明 ISCA 算法获得分割图像的速度更快，实时性更好，能够在保证分割质量的基础上减少运行时间。由以上分析可知，本文提出的基于 ISCA 的图像分割方法性能较好，能够胜任复杂图像的分割。

表 5　各算法指标对比

Table 5　Comparison of the indices obtained by each algorithm

图像	最大熵			运行时间（s）		
	本文算法	改进分水岭	模糊聚类	本文算法	改进分水岭	模糊聚类
Plant1	0.55482	0.21022	0.43235	2.382716	76.933121	116.095853
Plant2	0.77895	0.22682	0.67370	4.568249	89.217856	199.667319

5　结束语

本文提出了一种基于改进正余弦算法的多阈值图像分割方法，并利用该算法对图像分割的阈值选取过程加以优化。从伯克利图像的实验结果可知，该算法有效地提高了运行效率以及分割精度，表现出良好地适应性，很好地平衡了探索与开发能力，为图像的进一步处理奠定了基础。从植物冠层图像的分割实验可知，本文提出的算法能够完成复杂图像的分割。今后将会寻找更多有效的智能优化算法，并尝试与混沌搜索策略结合，将其应用到多阈值图像分割领域以提高算法的寻优精度。

参考文献：

[1] 薛志文，杨傲雷，费敏锐，等. 用于金属板图像分割的自适应阈值算法 [J]. 电子测量技术，2017, 40 (7): 85-89. (Xue Zhiwen, Yang Aolei, Fei Minyue, *et al.* Adaptive threshold algorithm for metal plate image segmentation [J]. Electronic Measurement Technology, 2007, 40 (7): 85-89.)

[2] 陈乔松，冉会琼，闫亚星，等. 基于数据场和水平集演化的图像分割 [J]. 重庆邮电大学学报:自然科学版 ，2018, 30 (2): 257-264. (Chen Qiaosong, Ran Huiqiong, Yan Yaxing, *et al.* Image segmentation based on data field and horizontal set evolution [J]. Journal of Chongqing University of Posts and Telecommunications:Natural Science , 2008, 30 (2): 257-264.)

[3] 杨章静，钱建军，黄璞，等. 基于 Gabor 变换的 GrabCut 纹理图像分割 [J]. 控制与决策，2016, 31 (1): 149-154. (Yang Zhangjing, Qian Jianjun, Huang Pu, *et al.* GrabCut texture image segmentation based on Gabor transformation [J]. Control and Decision-making, 2016, 31 (1): 149-154.)

[4] 张桂梅，王大雷. 结合 LPG&PCA 的中智学图像分割 [J]. 中国图象图形学报，2014, 19 (5): 693-700. (Zhang Guimei, Wang Dalei. Combined with LPG&PCA image segmentation [J]. Chinese Journal of Image and Graphics, 2014, 19 (5): 693-700.)

[5] 陈永艳，林丽华，田瑞. 基于自动识别理论的车牌定位算法研究 [J]. 内蒙古农业大学学报:社会科学版 ，2009, 11 (4): 119-121. (Chen Yongyan, Lin Lihua, Tian Rui. Research on license plate location algorithm based on automatic recognition theory [J]. Journal of Inner Mongolia Agricultural University:Social Science , 2009, 11 (4): 119-121.)

[6] 于玉海，林鸿飞，孟佳娜，等. 跨模态多标签生物医学图像分类建模识别 [J]. 中国图象图形学报，2018, 23 (6): 917-927. (Yu Yuhai, Lin Hongfei, Meng Jiana, *et al.* Classification modeling recognition of cross-modal multi-label biomedical images [J]. Chinese Journal of Image and Graphics, 1998, 23 (6): 917-927.)

[7] 胡加鑫，贾鹤鸣，邢致恺，等. 基于鲸鱼算法的森林火灾图像多阈值分割 [J]. 森林工程，2018, 34 (4): 1-6. (Hu Jiaxin, Jia Heming, Xing Zhikai, *et al.* Multi-threshold segmentation of forest fire images based

on whale algorithm [J]. Forest Engineering, 2018, 34 (4): 1-6.)
[8] Khairuzzaman A K M, Chaudhury S. Multilevel thresholding using grey wolf optimizer for image segmentation [J]. Expert Systems with Applications, 2017, 86: 64-76.
[9] Aziz M A E, A. Ewees A, Hassanien A E. Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation [J]. Expert Systems with Applications, 2017, 83: 242-256.
[10] Sarkar S, Das S, Chaudhuri S S. Multi-level thresholding with a decomposition-based multi-objective evolutionary algorithm for segmenting natural and medical images [J]. Applied Soft Computing, 2017, 50: 142-157.
[11] 柳新妮，马苗. 布谷鸟搜索算法在多阈值图像分割中的应用 [J]. 计算机工程, 2013, 39 (7): 274-278. (Liu Xinni, Ma Miao. The application of cuckoo search algorithm in multi-threshold image segmentation [J]. Computer Engineering, 2013, 39 (7): 274-278.)
[12] Mirjalili S. SCA: A Sine Cosine Algorithm for solving optimization problems [J]. Knowledge-Based Systems, 2016, 96: 120-133.
[13] 张海涛，程新文，熊红伟，等. 改进蜂群算法的图像阈值分割方法 [J]. 计算机应用研究, 2017, 34 (12): 3880-3884. (Zhang Haitao, Cheng Xinwen, Xiong Hongwei, *et al.* Image threshold segmentation method of improved bee swarm algorithm [J]. Application Eesearch of Computers, 2017, 34 (12): 3880-3884.)
[14] 杨陶，田怀文，刘晓敏，等. 基于双界线的 Otsu 阈值分割法及其快速算法 [J]. 计算机应用研究, 2016, 33 (12): 3872-3875. (Yang Tao, Tian Huaiwen, Liu Xiaomin, *et al.* Otsu thresholding segmentation based on double boundaries and its fast algorithm [J]. Application Research of Computers, 2016, 33 (12): 3872-3875.)
[15] 石雨豪，胡威旺，尚睿，等. 一种基于熵和聚类理论的图像阈值分割算法 [J]. 计算机应用研究, 2016, 33 (7): 2231-2235. (Shi Yuhao, Hu Weiwang, Shang Rui, *et al.* Image threshold segmentation algorithm based on entropy and clustering theory [J]. Application Research of Computers, 2016, 33 (7): 2231-2235.)
[16] 龙文，伍铁斌. 协调探索和开发能力的改进灰狼优化算法 [J]. 控制与决策, 2017, 32 (10): 1749-1757. (Long Wen, Wu Tiebin. Improved grey wolf optimization algorithm with coordinated exploration and development capability [J]. Control and Decision-making, 2007, 32 (10): 1749-1757.)
[17] 毕晓东，杨柏林. 基于多源特征与混沌算法的彩色图像加密算法 [J]. 控制工程, 2018, 25 (5): 855-862. (Bi Xiaodong, Yang Bolin. Color image encryption algorithm based on multi-source feature and chaos algorithm [J]. Control Engineering, 2008, 25 (5): 855-862.)
[18] 许栋，崔小欣，王田，等. 基于 Logistic 映射的混沌随机数发生器研究 [J]. 微电子学与计算机, 2016, 33 (2): 1-6. (Xu Dong, Cui Xiaoxin, Wang Tian, *et al.* Study on chaotic random number generator based on Logistic mapping [J]. Microelectronics and Computers, 2016, 33 (2): 1-6.)
[19] 曾艳阳，冯云霞，赵文涛. 基于 logistic 映射的自适应变尺度混沌粒子群算法 [J]. 系统仿真学报, 2017, 29 (10): 2241-2246. (Zeng Yanyang, Feng Yunxia, Zhao Wentao. An adaptive variable scale chaotic particle swarm algorithm based on logistic mapping [J]. Journal of System Simulation, 2007, 29 (10): 2241-2246.)
[20] 汪慎文，丁立新，谢承旺，等. 应用精英反向学习策略的混合差分演化算法 [J]. 武汉大学学报：理学版, 2013, 59 (2): 111-116. (Wang Shenwen, Ding Lixin, Xie Chengwang, *et al.* Hybrid differential evolution algorithm using elite reverse learning strategies [J]. Journal of Wuhan University (science) , 2013, 59 (2): 111-116.)
[21] 蒋刚毅，黄大江，王旭，等. 图像质量评价方法研究进展 [J]. 电子与信息学报, 2010, 32 (1): 219-226. (Jiang Gangyi, Huang Dajiang, Wang Xu, *et al.* Research progress of image quality evaluation methods [J]. Journal of Electronics and Information, 2010, 32 (1): 219-226)
[22] 冀甜甜，崔嘉，董新锋，等. 结合分水岭的纹理梯度各向异性图像分割 [J]. 中国图象图形学报, 2017, 22 (7): 926-934. (Ji Tiantian, Cui Jia, Dong Xinfeng, *et al.* Texture gradient anisotropy image segmentation combined with watershed [J]. Journal of Image and Graphics, 2017, 22 (7): 926-934)
[23] 王展，杜平安，李杨，等. 基于 FCM 聚类的示温漆图像分割算法 [J]. 航空动力学报, 2018, 33 (3): 604-610. (Wang Zhan, Du Pingan, Li Yang, *et al.* Image segmentation algorithm based on FCM clustering [J]. Journal of Aerospace Power, 2018, 33 (3): 604-610)
[24] 胡学刚，严思奇. 基于 FCM 聚类的图像分割算法 [J]. 计算机工程与设计, 2018, 39 (1): 159-164. (Hu Xuegang, Yan Siqi. Image segmentation algorithm based on FCM clustering [J]. Computer Engineering and Design, 2018, 39 (1): 159-164)